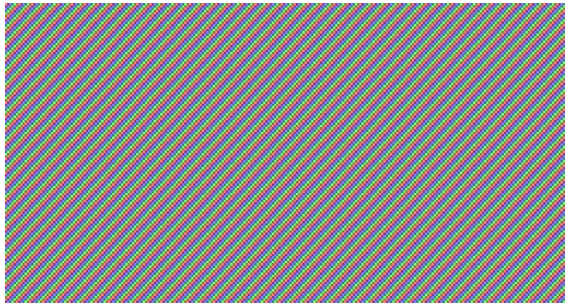
- JPEG is a lossy image compression method. The effectiveness of the DCT transform coding method in JPEG relies on three major observations:
- Observation-1:
  - Useful image contents change relatively slowly across the image— that is, it is unusual for intensity values to vary widely several times in a small area—for example, in an 8 × 8 image block.
  - ***Spatial frequency*** indicates how many times pixel values change across an image block.
  - The DCT formalizes this notion with a measure of how much the image contents change in relation to the number of cycles of a cosine wave per block.

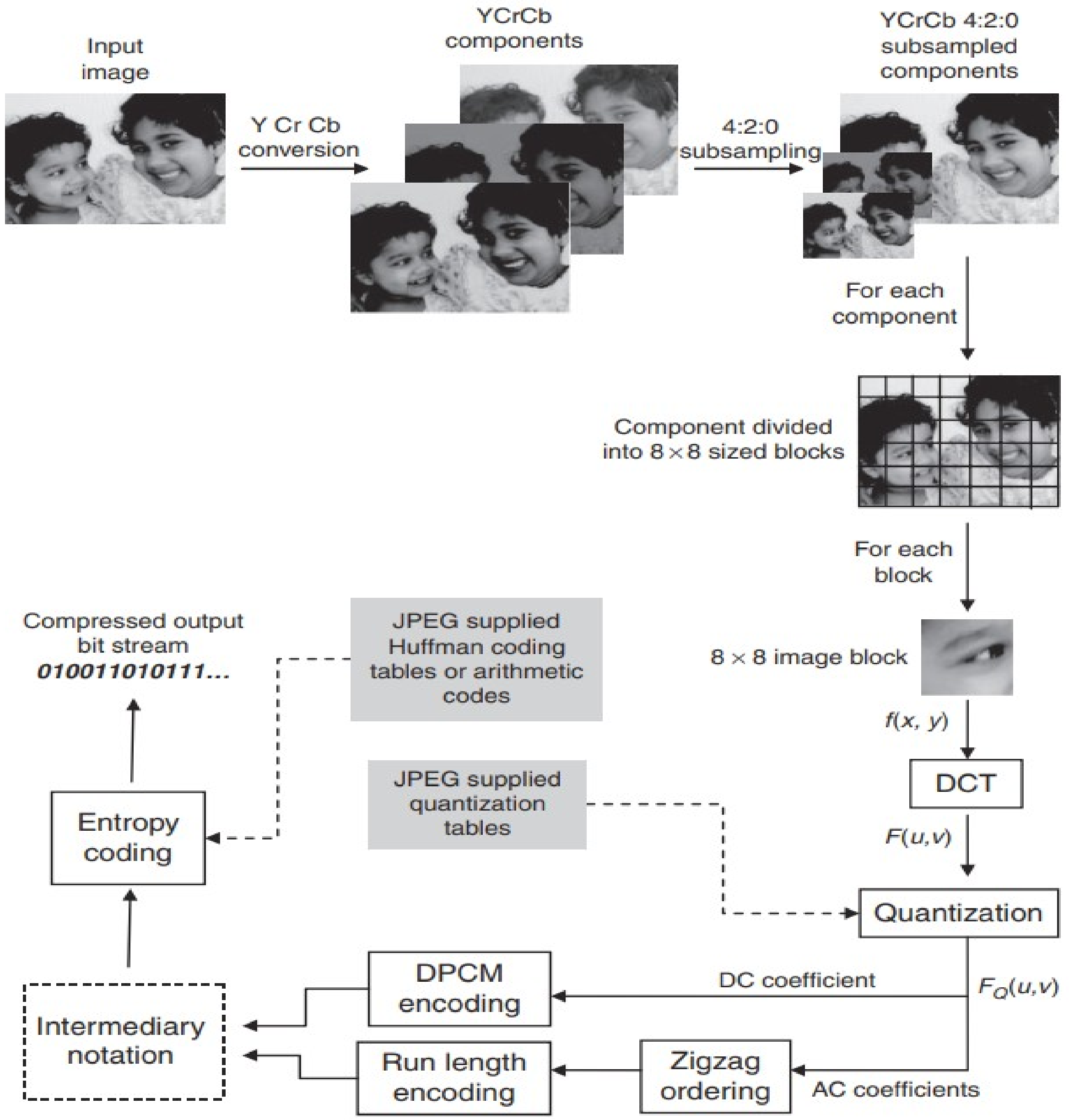
- Observation-2:
  - Psychophysical experiments suggest that humans are much less likely to notice the loss of very high-spatial frequency components than lower frequency components.
- Observation-3:
  - Visual acuity (accuracy in distinguishing closely spaced lines) is much greater for gray (“black and white”) than for color.

Low spatial frequency



High spatial frequency

JPEG Compression pipeline



- The steps followed in JPEG compression are as follows.
- ***Step 1:*** Any image can be taken as input, but is always first converted to the YCrCb format to decouple the image chrominance from the luminance.
- ***Step 2:*** The YCrCb representation undergoes a 4:2:0 subsampling, where the chrominance channels Cr and Cb are subsampled to one-fourth the original size.
- ***Step 3:*** Each channel (Y, Cr, and Cb) is processed independently. Each channel is divided into 8×8 blocks.
  - On the average, the 8×8 size seems to be the most optimal area for spatial and spectral correlation that the DCT quantization can exploit. Smaller sizes increase the number of blocks in an image, and larger sizes reduce the correlation seen among pixels.
  - If the image width (or height) is not a multiple of 8×8, the boundary blocks get padded with zeros to attain the required size. The blocks are processed independently.

- ***Step 4:*** Each 8×8 block (for all the channels) undergoes a DCT transformation, which takes the image samples f(x,y) and computes frequency coefficients F(u,v).
  - The first coefficient F(0, 0), is normally the highest, and is called the ***DC coefficient***.
  - This special status for the DC coefficient is deliberate because most of the energy in natural photographs is concentrated among the lowest frequencies. The remaining coefficients are called ***AC coefficients***.
- ***Step 5:*** The DCT coefficients F(u,v) are quantized using a quantization table supplied by JPEG.
  - Each number at position (u,v) gives the quantization interval size for the corresponding F(u,v) value.
  - The quantization table values might appear random, but, in fact, they are based on experimental evaluations with human subjects, which have shown that low frequencies are dominant in images, and the human visual system is more sensitive to loss in the low-frequency range.
  - Correspondingly, the numbers in the low-frequency area (upper-left corner of the table) are smaller and increase as you move toward the high-frequency coefficients in the other three corners.

178	187	183	175	178	177	150	183
191	174	171	182	176	171	170	188
199	153	128	177	171	167	173	183
195	178	158	167	167	165	166	177
190	186	158	155	159	164	158	178
194	184	137	148	157	158	150	173
200	194	148	151	161	155	148	167
200	195	172	159	159	152	156	154

Pixel values $f(x, y)$

1359	46	61	26	38	−21	−5	−18
31	−35	−25	−11	13	10	12	−3
13	20	−17	−14	−11	−7	6	5
−5	5	2	−8	−11	−26	8	−4
10	15	−10	−16	−21	−7	8	7
−6	1	0	7	5	−7	−1	−3
−13	−8	1	10	8	4	−3	−4
−5	−5	−2	5	5	0	0	−3

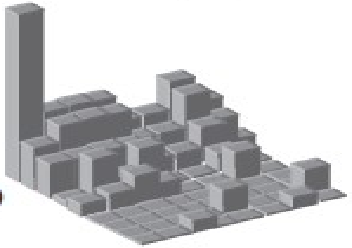
DCT values $F(u,v)$

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Quantization table

$f(x, y)$

DCT



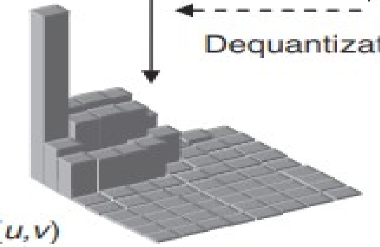
$F(u,v)$

Quantization

$F_Q(u,v)$

85	4	6	2	2	−1	0	0
3	−3	−2	−1	1	0	0	0
1	2	−1	−1	0	0	0	0
0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Dequantization



$\hat{F}(u,v)$

Inverse DCT

$\hat{f}(x, y)$

192	185	178	152	193	162	155	190
181	172	162	154	187	164	159	192
183	168	150	158	181	167	162	190
198	177	150	155	177	169	161	182
202	180	148	148	171	167	159	175
193	176	145	141	162	162	156	170
195	184	155	145	159	156	150	164
209	200	170	154	160	153	144	156

1360	44	60	32	48	−40	0	0
36	−36	−28	−19	26	0	0	0
14	26	−16	−24	0	0	0	0
0	0	0	0	0	0	0	0
18	22	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

- Using this table, $F_Q(u,v)$ is computed as

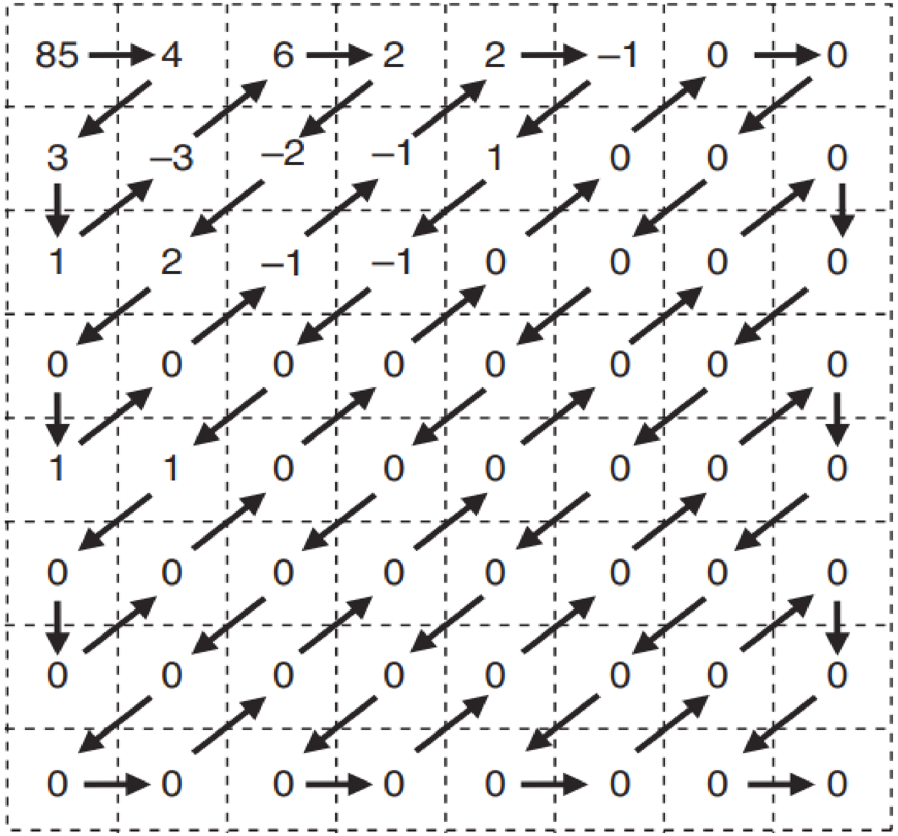
$$F_Q(u,v) = \left\lceil \frac{F(u,v)}{Q(u,v)} \right\rfloor$$

where Q(u, v) is the value in the quantization table.

- After quantization, almost all the high-frequency FQ(u,v) are zero, while a few low frequency values remain.

- ***Step 6:*** The quantized coefficients FQ(u, v) are then encoded into an intermediary pattern.
  - DC FQ(0,0), which normally corresponds to the highest energy, is treated differently when compared with the other higher-frequency coefficients, called AC coefficients.
  - The DC coefficients of the blocks are encoded using differential pulse code modulation.
  - The AC coefficients are first scanned in a zigzag order

Zigzag ordering of quantized AC coefficients. The resulting sequence has a much longer run of zeros and, hence, can be more efficiently entropy coded.



DC coefficient = 85

AC coefficient stream

4 3 1 -3 6 2 -2 2 0 1 0 -1 -1 2
-1 1 -1 0 1 0 0 0 0 0 0 0 0 0 ...

- The zigzag ordering produces a longer run of zeros towards the end of the scan because the high-frequency coefficients appear at the tail end. This produces a lower entropy of scanned AC coefficients, which are run length encoded.
- Both the DPCM codes of DC coefficients and the run length coded AC coefficients produce an intermediary representation.

- ***Step 7:*** The next step is to entropy code the run of DC and AC coefficients. Prior to entropy coding, these coefficients are converted into intermediary representations.
  - For the representation of the DC coefficient, it is first DPCM coded with the previous block's DC value and the difference is represented as a 2-tuple, which shows the size in bits used to encode the DPCM difference and the amplitude of the DPCM difference.
  - In our example, the quantized DC value of the block is 85. Assuming that the DC value of the previous block was 82, the DPCM difference is 3, which needs two bits to encode. Therefore, the intermediary notation of the DC coefficient is <2><3>.

DC coefficient representation:

symbol -1 <SIZE>

symbol -2 <AMPLITUDE>

AC coefficient representation:

symbol -1 <RUNLENGTH, SIZE>

symbol -2 <AMPLITUDE>

Intermediary stream

<2><3> <0,3><4> <0,2><3> <0,1><1> <0,2><-3> <0,3><6>
<0,2><2> <0,2><-2> <0,2><2> <1,1><1> <1,1><-1> <0,1><-1>
<0,2><2> <0,1><-1> <0,1><1> <0,1><-1> <1,1><1> EOB

- For AC coefficients, the intermediary notation is used only for the nonzero AC coefficients.
- Each nonzero AC coefficient is again represented by two symbols.
  - The first symbol here represents two pieces of information—runlength and size. The runlength is the number of zeros in the run that precede the non-zero coefficient in the zigzag sequence and the size is the number of bits used to encode the amplitude of the non-zero AC coefficient.
  - The second symbol encodes the amplitude of the nonzero AC coefficient.
  - In our example, the first nonzero AC coefficient has a value of 4 and there are no zeros preceding it, so the run length is 0. The amplitude of 4 needs 3 bits (as per the JPEG codes) to encode. Therefore, the intermediary representation is <0,3><4>.

- ***Step 8:*** The intermediary representations are then entropy coded using codes supplied by the JPEG organization.
- The first symbol in the intermediary representation for both DC and AC coefficients is encoded using Huffman coding.
- The second symbol, which is the amplitude, is encoded using variable length integer code.
- To save bits, RUNLENGTH and SIZE are allocated only 4 bits each and squeezed into a single byte.
- The 4-bit RUNLENGTH can represent only zero-runs of length 0 to 15. Occasionally, the zero run-length exceeds 15; then a special extension code, (15, 0), is used for Symbol 1

Baseline entropy coding details—size category

Size	Amplitude
1	$-1, 1$
2	$3, -2, 2, 3$
3	$-7 .. -4, 4 .. 7$
4	$-15 .. -8, 8 .. 15$
.	.
.	.
.	.
10	$-1023 .. -512, 512 .. 1023$

Binary Stream:
0111110010001110010100
1001100110010101101 1
111000001100000010001
111010

Intermediary stream
<2><3> <0,3><4> <0,2><3> <0,1><1> <0,2><-3> <0,3><6>
<0,2><2> <0,2><-2> <0,2><2> <1,1><1> <1,1><-1> <0,1><-1>
<0,2><2> <0,1><-1> <0,1><1> <0,1><-1> <1,1><1> EOB

Intermediary symbol	Binary representation of first symbol (prefixed Huffman Codes)	Binary representation of second symbol (non-prefixed variable integer codes)
<2> <3>	011	11
<0,3> <4>	100	100
<0,2> <3>	01	11
<0,1> <1>	00	1
<0,2> <−3>	01	00
<0,3> <6>	100	110
<0,2> <2>	01	10
<0,2> <−2>	01	01
<0,2> <2>	01	10
<1,1> <1>	11	1
<1,1> <−1>	11	0
<0,1> <−1>	00	0
<0,2> <2>	01	10
<0,1> <−1>	00	0
<0,1> <1>	00	1
<0,1> <−1>	00	0
<1,1> <1>	11	1
EOB	1010	

JPEG compression results. The upper-left image shows the original at 24 bits per pixel. The remaining three images show the reconstructed outputs after JPEG compression at different compression ratios. The blocky artifacts increase at lower bit rates.

Original—24 bits per pixel

Compressed—2 bits per pixel

Compressed—0.5 bits per pixel

Compressed—0.15 bits per pixel